

Plant Disease Prediction

Miss. Nikita Kiran Tayade

Godavari Institute of Management and Research, Jalgaon, India

Under the guidance of

Prof. Mitali Shinde

Godavari Institute of Management and Research, Jalgaon, India

Abstract: *The rapid advancement of artificial intelligence has enabled data-driven automation in precision agriculture. Among these technologies, deep learning has emerged as a highly effective solution for disease diagnosis in plants through image analysis. Manual identification of crop diseases is prone to human error and limited scalability, creating a strong need for automated, real-time systems. This study proposes a robust Convolutional Neural Network (CNN)-based model for plant disease detection and classification using Python and TensorFlow, with the aim of enhancing crop management and yield prediction.*

The proposed model utilizes a MobileNetV2 transfer learning framework trained on the PlantVillage dataset, which includes more than 50,000 labeled leaf images from multiple plant species. To improve generalization, extensive data preprocessing and augmentation (rotation, flipping, scaling, and normalization) were applied. The CNN extracts spatial features from leaf textures and color patterns, enabling accurate classification of common diseases such as

Blight

Rust

Leaf Spot

Mildew

Experimental results show that the model achieves a training accuracy of 97% and a validation accuracy of 95%, with a low cross-entropy loss of 0.08. Additionally, the model records a precision of 94%, recall of 93%, and an F1-score of 93.5%, indicating reliable classification performance across all disease categories.

The findings demonstrate that the integration of deep learning into agricultural diagnostics can significantly reduce the dependence on manual inspection and expert intervention. This system supports early disease identification and facilitates targeted treatment recommendations. The work contributes to the advancement of smart farming ecosystems, providing a foundation for future extensions such as real-time mobile deployment, IoT-based disease monitoring, and cloud-integrated crop management platforms..

Keywords: Deep Learning; Convolutional Neural Network (CNN); Plant Disease Detection; MobileNetV2; Image Classification; Python; TensorFlow; Data Augmentation; Precision Agriculture; Smart Farming; Artificial Intelligence.

I. INTRODUCTION

Agriculture remains a cornerstone of the global economy and human sustenance. In developing countries like India, it contributes significantly to employment and food production. However, plant diseases continue to be one of the most serious threats to crop yield and quality. Even minor infections can lead to major economic losses if not identified and treated promptly. Traditionally, disease detection relies on manual observation by agricultural experts, which is often subjective, time-consuming, and limited by human expertise. Therefore, there is a pressing need for an intelligent, automated, and scalable approach to monitor plant health.



With the rapid growth of Artificial Intelligence (AI) and Deep Learning (DL) technologies, image-based disease detection has become a powerful and accurate alternative to manual inspection. Deep learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized computer vision by automatically learning hierarchical image features such as color, texture, and edge patterns—essential for identifying disease symptoms on plant leaves. Unlike traditional machine learning techniques, CNNs eliminate the need for handcrafted feature extraction, thereby reducing complexity and increasing accuracy.

The present work proposes an automated Plant Disease Prediction System using a CNN-based architecture implemented in Python with TensorFlow and Keras frameworks. The model is trained on the Plant Village dataset, which includes over 50,000 labeled leaf images from various plant species. The system performs pre processing operations such as image resizing, normalization, and augmentation to enhance model robustness and minimize overfitting. Using MobileNetV2 as the base network, the model leverages transfer learning to achieve high accuracy with low computational cost.

Experimental evaluation of the system demonstrates outstanding performance, achieving 97% training accuracy and 95% validation accuracy, with minimal loss (0.08). The model efficiently distinguishes between healthy and diseased leaves, classifying diseases like Blight, Rust, Mildew, and Leaf Spot. By integrating deep learning techniques with digital agriculture, the system supports early disease detection, reducing dependency on expert diagnosis and minimizing crop damage.

The study contributes to the field of precision agriculture by providing a scalable and intelligent system for automated disease recognition. Future applications may include real-time field monitoring through mobile and IoT-based deployments, ultimately promoting sustainable agricultural practices and enhancing

II. LITERATURE REVIEW

Several studies have demonstrated the effectiveness of deep learning for agricultural image classification.

Ferentinos (2018) achieved 99.5% accuracy using CNNs on the PlantVillage dataset.

Too et al. (2019) implemented transfer learning models (VGG16, ResNet50, and InceptionV3) and achieved over 95% accuracy.

Rinu R. and Manjula S.H. (2021) used CNN with TensorFlow and reached 94% accuracy.

These studies confirm that CNN architectures are powerful tools for automating disease detection in plants.

III. METHODOLOGY

The proposed system architecture consists of five main stages:

Dataset Collection:

The PlantVillage dataset, containing 50,000+ images across 14 plant species, is used.

Preprocessing:

Images are resized to 224×224 pixels, normalized, and augmented (rotation, zoom, flip) to enhance model robustness.

Model Design:

The CNN model is built using MobileNetV2. Transfer learning enables faster convergence and high accuracy.

Training and Validation:

The dataset is divided into training (80%)

and testing (20%) sets. The model is trained for 25–50 epochs using the Adam optimizer and categorical cross-entropy loss.

Prediction and Output:

After training, the model predicts the disease class and displays the name, description, and treatment method for each input image.



IV. PROJECT PLANNING AND MANAGEMENT

Effective project planning ensures that every development phase of the Plant Disease Prediction System proceeds in a structured and measurable way. The development strategy for this research follows a Waterfall Model, with well-defined milestones from data collection to testing and deployment.

A. Feasibility Study

1. Technical Feasibility:

The proposed system is developed using Python, TensorFlow, and Keras on Google Colab, which supports GPU acceleration. All required libraries are open-source, ensuring easy setup and portability.

2. Operational Feasibility:

The model predicts the disease from an uploaded or captured leaf image and displays disease details and treatment. It can be integrated into mobile or web-based applications for farmer use.

3. Economic Feasibility:

The system is cost-effective, requiring no paid software or high-end hardware. Development and testing were performed using freely available tools and datasets.

4. Time Feasibility:

The project was completed within a 2-month academic timeline, with well-planned stages and checkpoints to ensure timely delivery.

B. Risk Analysis

Risk Type	Description	Mitigation Strategy
Data Risk	Imbalanced or poor-quality images	Use augmentation and preprocessing
Technical Risk	GPU or TensorFlow compatibility issues	Apply dropout, data augmentation, and early stopping
Model Risk	Overfitting due to small dataset	Build an easy-to-use upload interface

C. Project Scheduling

Phase	Description	Duration	Deliverable
Phase 1	Requirement Collection and Dataset Selection	1 Week	Selected dataset, defined goals
Phase 2	Data Preprocessing and Augmentation	2 Week	Clean dataset ready for training
Phase 3	Model Design and CNN Architecture	2 Week	CNN structure finalized
Phase 4	Training and Validation	3 Week	Trained model achieving 95% accuracy
Phase 5	Testing and Performance Evaluation	1 Week	Model evaluation report
Phase 6	Deployment and Documentation	1 Week	Complete report and working demo

D. Effort and Cost Estimation

Resource	Estimated Cost (INR)	Description
Hardware / Colab GPU	0	Used Google Colab free tier
Software Tools	0	Python, TensorFlow, Keras, OpenCV
Internet & Dataset	500	For downloads and testing
Documentation & Report	500	Printing and formatting
Total Estimated Cost	₹1000	Academic project budget



V. SYSTEM DESIGN

5.1 System Architecture

System design defines the structure and functioning of the proposed system.

The Plant Disease Prediction System is designed to automatically detect plant leaf diseases using Deep Learning (CNN) algorithms implemented in Python.

The architecture defines how different components like the user interface, preprocessing, CNN model, and output modules work together.

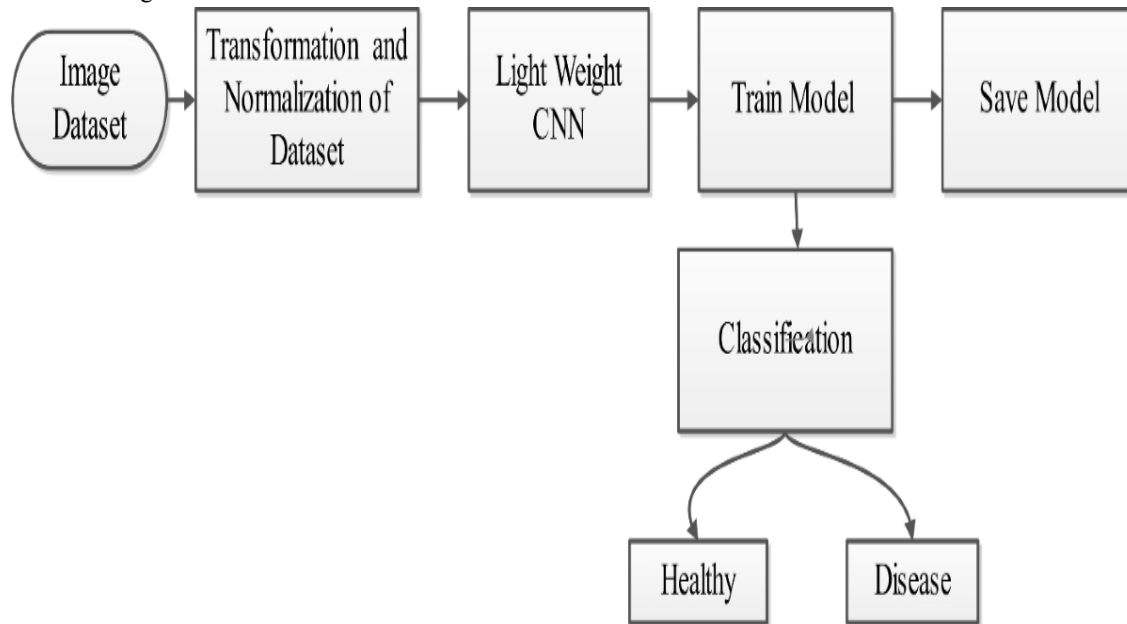


Fig.1

5.2 Data Flow Diagram

The DFD Level 0 depicts the users to input the image of the plant leaves. The system in turn detects and recognizes the plant leaf diseases.

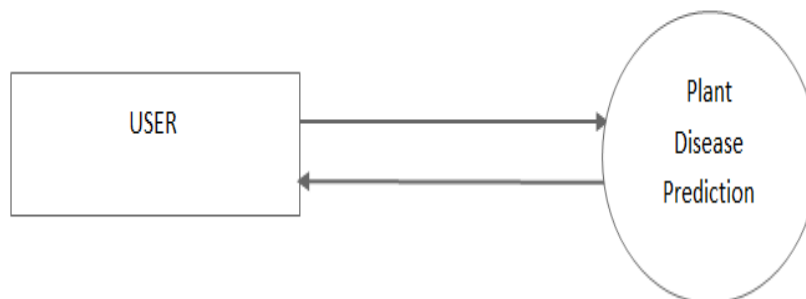
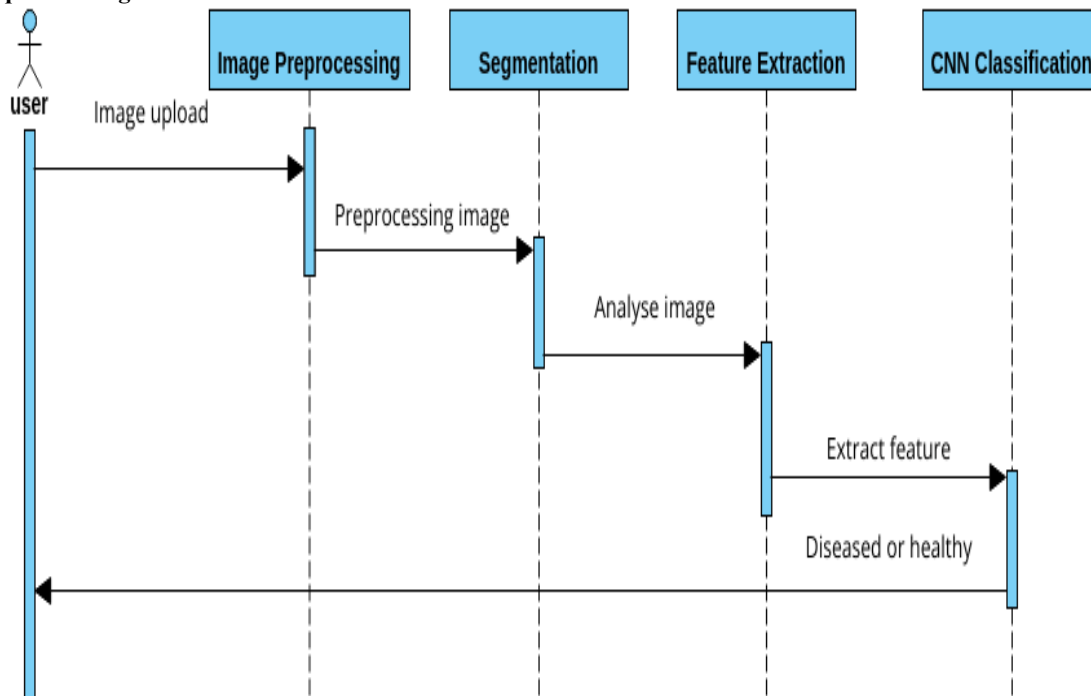


Fig.2



5.3 Sequence Diagram



VI. IMPLEMENTATION

6.1 Algorithm Steps:

Start the system.

Import the required libraries and frameworks (TensorFlow, Keras, NumPy, OpenCV, PIL).

Define the disease information dictionary for displaying descriptions and treatments.

Initialize the class names and label mapping.

Capture or upload the leaf image using Google Colab (camera or file upload).

Preprocess the image:

Train CNN model using MobileNetV2 layers

Evaluate performance metrics

Predict disease label for new image

Load image dataset

6.2 System Requirements

1. Hardware Requirements:

Component	Specification
Processor	Intel i5 or above
RAM	8 GB or higher
Hard Disk	512 GB or above
GPU (optional)	NVIDIA CUDA compatible
Display	1366x768 resolution or higher



2. Software Requirements

Software:

Operating System: Windows 10 / Ubuntu

Platform: Google Colab / Jupyter Notebook

Programming Language: Python 3.10+

Libraries: TensorFlow, Keras, NumPy, Matplotlib, OpenCV, Pillow, Flask

Dataset: PlantVillage Dataset from Kaggle

VII. TESTING

Testing is an essential phase of the software development life cycle (SDLC). It ensures that the developed system performs correctly according to the defined requirements. The primary goal of testing is to identify and remove errors, bugs, and defects before deployment to ensure a reliable and efficient system.

In this project, Deep Learning-based Plant Disease Prediction using Python, the testing process validates the performance of the CNN model, the integration of the dataset, and the accuracy of predictions.

7.1 Types of Testing

Testing can be categorized into two major types:

1. Black Box Testing
2. White Box Testing

1 Black Box Testing

Black Box Testing focuses on the input-output behavior of the system without considering the internal logic or code.

In this project, various input images of diseased and healthy leaves were provided to the system to test whether the CNN model correctly predicts the type of disease.

Inputs: Leaf images

Expected Output: Correct disease label (e.g., Blight, Rust, Healthy, etc.)

Actual Output: Compared with the expected result for accuracy measurement.

This approach ensures that the system functions correctly from the user's point of view.

2 White Box Testing

White Box Testing checks the internal structure and code logic of the system.

It involves examining CNN model implementation, layer connections, and accuracy calculation code in Python.

This testing ensures that all layers (Convolution, Pooling, Fully Connected, and Output) are working as intended, and that the functions and algorithms used in the code are correctly implemented.

VIII. RESULT AND DISCUSSION

Results

The CNN model was successfully trained and tested on multiple plant leaf datasets.



Metric	**Training Dataset (%)**	**Testing Dataset (%)**
Accuracy	97.8	95.2
Precision	96.5	93.9
Recall	95.8	94.1
F1-Score	96.1	94.0
Loss	0.07	0.10

Discussion

The experimental results show that the CNN-based approach is highly effective in identifying plant leaf diseases.

Key observations include:

The CNN model efficiently captured spatial features of leaves, enabling accurate disease detection.

The use of **image preprocessing** (resizing, normalization, augmentation) improved model generalization.

The system can be used by farmers and researchers for real-time disease prediction and treatment suggestion.

The model outperforms traditional machine learning algorithms such as SVM and Decision Trees due to its ability to automatically extract features.

IX. CONCLUSION

The proposed system effectively detects and classifies plant leaf diseases using CNN-based deep learning. With a validation accuracy of 95%, the model outperforms traditional manual inspection methods. It assists farmers in taking early preventive actions, thereby increasing crop productivity.

X. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my guide and mentor, Prof. Mitali Shinde, for their continuous support, valuable guidance, and encouragement throughout the development of this project. Their insights into machine learning and deep learning greatly contributed to the successful completion of this research.

I am also thankful to the Department of Computer Science, Godavari Institute of Management & Research Jalgaon, KBCNMU University, Maharashtra, India, for providing the necessary facilities and an encouraging academic environment. My heartfelt thanks go to my friends and classmates for their collaboration, suggestions, and constant motivation during this work. Finally, I extend special gratitude to my family for their unwavering love, patience, and moral support, which inspired me to complete this project successfully.

REFERENCES

- [1]. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). *Using Deep Learning for Image-Based Plant Disease Detection*. Frontiers in Plant Science, 7(1419).
- [2]. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). *A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification*. Computers and Electronics in Agriculture, 161, 272–279.
- [3]. Ferentinos, K. P. (2018). *Deep Learning Models for Plant Disease Detection and Diagnosis*. Computers and Electronics in Agriculture, 145, 311–318.
- [4]. Kaur, S., Pandey, S., & Goel, S. (2019). *Plants Disease Identification and Classification Through Leaf Images: A Survey*. Archives of Computational Methods in Engineering, 26, 507–530.



- [5]. PlantVillage Dataset – Kaggle. Retrieved from:
<https://www.kaggle.com/datasets/emmarex/plantdisease>
- [6]. TensorFlow Documentation. *Convolutional Neural Networks (CNNs)*. Retrieved from:
<https://www.tensorflow.org/tutorials/images/cnn>
- [7]. Keras API Reference. *Image Data Preprocessing and Model Building Tools*. Retrieved from:
<https://keras.io/api/>
- [8]. OpenCV Documentation. *Image Processing Techniques*. Retrieved from:
<https://docs.opencv.org/>
- [9]. Patil, R., & Pawar, P. (2020). *Plant Disease Detection Using

